

REMARKS

This paper is responsive to a Final Office Action dated January 26, 2005. Prior to this response claims 1-17 were pending. Claims 1-17 remain pending.

In Section 4 of the Office Action claims 1-17 have been rejected under 35 U.S.C. 112, second paragraph as being indefinite. The Office Action states that "(c)laims 1, 7, and 12 which recite newly limitation "defining device user interface control" in response to the prior rejection do not make the scopes of the Claims clear." More specifically, the Office Action states that the claim element "retrieving virtual key information" does not reflect what is set forth in the claim preamble. In summary, the Office Action states that the claim fails to particularly point out and claim the subject matter. This rejection is traversed as follows.

The second paragraph of 35 U.S.C. 112 states that "(t)he patent specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter of the invention." As noted in MPEP 2171, the second paragraph of U.S.C. 112, sets forth two requirements; that the claim sets forth the subject matter of the invention, and that the claims set forth the metes and bounds of the subject matter.

As noted in MPEP 2173.02, "(i)n reviewing a claim for compliance with 35 U.S.C. 112, second paragraph, the examiner must consider the claim as a whole to determine whether the claim apprises one of ordinary skill in the art of its scope, and therefore, serves the notice function required by 35 U.S.C. 112, second paragraph..." *Solomon v.*

Kimberly-Clark Corp., 216 F.3d 1372, 1379, 55 USPQ2d 1279, 1283 (Fed. Cir. 2000).

Even more to the point, MPEP 2173.05(e) states that "(t)he mere fact that the body of the claim recites additional elements which do not appear in the claim's preamble does not render the claim indefinite under 35 U.S.C. 112, second paragraph *In re Larson*, No. 01-1092 (Fed. Cir. May 9, 2001).

In summary, the Applicant respectfully submits that there is no requirement that the preamble of a claim list every claim element recited in the body of the claim. Further, the Applicant submits that a claim cannot be judged as indefinite merely for what is recited, or not recited in the preamble. Finally, the Applicant submits that, read as a whole, claims 1, 7, and 12 definitely claim the subject matter of the invention, and the Applicant requests that the rejection be removed.

In Section 7 of the Office Action claims 1-17 have been rejected under 35 U.S.C. 102(a) as being anticipated by HAVi Specification Version 1.1, 5-2001 ("HAVi"). With respect to claim 1, the Office Action states that HAVi discloses the retrieval of virtual key information in response to accessing a JAR file (Section 1.3, page 5, 2.5.2, and 2.7.2). Sections 2.9.2 and Section 8.3.2.5 (page 429) are referenced with respect to retrieving virtual key information.

With respect to claim 7, the Office Action states that HAVi discloses the retrieval of virtual key information in response to accessing a ResourceBundle. The ResourceBundle is referenced with Sections 1.3, 2.7.2, 2.5.2, 2.9.2, 8.3.2.4, and 8.3.2.5. With respect to claim 12, the Office Action states that HAVi discloses the retrieval of virtual key information in response to accessing a mapped memory, referencing Sections 1.3,

2.7.2, 2.5.2, 2.9.2, 7.4, 8.1, 8.7, 8.8, 3.10.1, 9.4, 9.5, and 8.3.2.5. This rejection is traversed as follows.

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

With respect to the HAVi Sections cited in the Office Action against claim 1:

Section 1.3 is a chart of terminology, which at page 5, line 10, lists, "HAVi Level 2 interoperability".

Section 2.5.2 states that the L2 UI is based upon JAVA AWT 1.1.

Section 2.7.2 describes Level 2 interoperability.

Section 2.9.2 describes Signature Verification.

Section 8.3.2.5 (page 429) states that an event can have a representation as a string, color, or symbol, which can be determined by calling *getString*, *getColor*, and *getSymbol*, respectively. This methodology permits the definition of a device button. None of the above mentioned HAVi Sections teach how the virtual key information is to be stored or retrieved from memory. In contrast, claim 1 recites the retrieval of virtual key information in response to accessing a JAR file stored in memory.

With respect to claim 1, the *Response to Arguments* Section states that Section 2.5.2, 7.2.2, and 3.10.1 disclose the accessing of virtual key information from JAR files. In response, the Applicant notes that the cited portions of the HAVi specification do not describe the retrieval of

virtual key information, in response to accessing a JAR file, as recited in claim 1.

Section 2.5.2 describes a Level 2 user interface that can be used to support display screen functions, alpha blending, remote control inputs, and support for visual interface components. Nowhere in this section is there a description of accessing a JAR file to retrieve virtual key information. Section 7.2.2, at page 394, describes the packages and classes that may be used in DCM and Application Module code units. Nowhere in this section is there a description of accessing a JAR file to retrieve virtual key information. Fig. 25 (Section 3.10.1), at page 89, describes the retrieval and use of digital signatures. Section 3.10.1 does not include a description of accessing a JAR file to retrieve virtual key information.

In summary, the cited HAVi specifications do not describe accessing a JAR file to retrieve virtual key information. These cited sections, and other uncited sections of the HAVi specification are general implementation guidelines. Claim 1, while operating in the context of the HAVi specification, is a narrower invention that includes limitations that cannot be found in the HAVi specification.

The *Response to Arguments* Section of the Office Action states (page 3, second-last paragraph) that, "...the above Applicant's assertions tend to short, not to extend to the whole HAVi specification which incorporates the Claim's limitations. Respectfully, the whole consideration of Havi Specification would be required."

The Applicant assumes that the above-quoted section of the Office Action is intended to mean that the Applicant's claims must be read in the context of the entire HAVi specification, not just the specific HAVi

specification sections listed in the Office Action. However, the Applicant is unaware of any other sections, or combinations in the HAVi specification that can be used to support the Examiner's assertions.

In short, HAVi does not give any guidance as to virtual key information storage format. Claim 1 recites that the storage format is a JAR file. A JAR file may be stored in an EEPROM for example, making it easy to change, update, or modify.

With respect to claim 7, the cited HAVi specification Sections include:

Section 1.3, as noted above, is a chart of terminology, which at page 5, line 10, lists, "HAVi Level 2 interoperability".

Section 2.5.2 states that the L2 UI is based upon JAVA AWT 1.1.

Section 2.7.2 describes Level 2 interoperability.

Section 2.9.2 describes Signature Verification.

Section 8.3.2.4 states that there are three classes available to determine if a device is available.

Section 8.3.2.5 (page 429) of the HAVi specification states that an event can have a representation as a string, color, or symbol, which can be determined by calling *getString*, *getColor*, and *getSymbol*, respectively.

Claim 7 recites the retrieval of virtual key information in response to accessing a *ResourceBundle*. None of the above-mentioned HAVi Sections teach the retrieval of virtual key information in response to accessing a *ResourceBundle*.

The *Response to Arguments* Section of the Office Action states that that Section 8.3.2.5 of the HAVi specification describes input elements such as String, Color, and Symbol. The *Response to Arguments* Section continues, stating that 8.3.2.5 and Section 2.9.2 describe 6 keys that may be obtained from calling "getColor".

In response, the Applicant notes that Section 8.3.2.4 states that there are three classes available to determine if a device is available. Section 8.3.2.5 states that an event (device button) can have a representation such as a string, color, or symbol, which can be determined by calling *getString*, *getColor*, and *getSymbol*, respectively. However, nowhere in these sections is there a description of accessing a ResourceBundle to retrieve virtual key information. Section 2.9.2 of the HAVi specification describes a signature verification process. Nowhere in this section is there a description of accessing a ResourceBundle to retrieve virtual key information.

Claim 7, while operating in the context of the HAVi specification, is a narrower invention that includes limitations that cannot be found in the HAVi specification.

With respect to claim 12, the cited HAVi Specification Sections are:

Section 1.3 is a chart of terminology, which at page 5, line 10, lists, "HAVi Level 2 interoperability".

Section 2.5.2 states that the L2 UI is based upon JAVA AWT 1.1.

Section 2.7.2 describes Level 2 interoperability.

Section 2.9.2 describes Signature Verification.

Section 7.4 states that JAVA code units are entities for uploading, and that the format of a JAVA code unit is the JAR format. Details are given of DCM, AMC, and Havlet code units.

Section 8.1 describes the HAVi user interface design, using a subset of AWT as defined in JAVA 1.1.

Section 8.7 describes a general approach to error behavior.

Section 8.8 is a list of constants.

Fig. 25 (Section 3.10.1), at page 89, describes the retrieval and use of digital signatures.

Section 9.4 presents a list of defined HAVi key values.

Section 9.5 lists HAVi and non-HAVi ROM requirements.

Section 8.3.2.5 (page 429) of the HAVi specification states that an event can have a representation as a string, color, or symbol, which can be determined by calling *getString*, *getColor*, and *getSymbol*, respectively.

Claim 12 recites the retrieval of virtual key information in response using a JNI to access a mapped memory. None of the above mentioned HAVi Sections teach the use of a JNI to access mapped memory, to retrieve virtual key information.

The *Response to Arguments* Section of the Office Action states that the Level 2 UI provides access to keys to a code like a JAR file that resides in memory (Section 7.4). The Office Action states that the format of a Java Code unit is the Java archive or JAR format. The Office Action states that page 89 shows HAVi uploading/obtaining keys, that Sections 9.4 and 9.5 show key representations, and that Section 9.7 discusses means for accessing mapped memory where JAR native code resides.

In response, the Applicant notes that Section 7.4 of the HAVi specification describes that JAVA code units are entities for uploading, and that the format of a JAVA code unit is the JAR format. Nowhere in this section is there a description of the retrieval of virtual key information in response using a JNI to access a mapped memory.

Fig. 25 (Section 3.10.1), at page 89, describes the retrieval and use of digital signatures. Section 3.10.1 does not include a description of the retrieval of virtual key information in response using a JNI to access a mapped memory. Section 9.4 presents a list of defined HAVi key values. Section 9.5 lists HAVi and nonHAVi ROM requirements. Neither of these sections describes the retrieval of virtual key information in response using a JNI to access a mapped memory. Section 9.7 of the HAVi specification describes the GUID and Bus_Info_Block. This section does not describe the retrieval of virtual key information in response using a JNI to access a mapped memory.

In general, the HAVi specification does not explicitly describe claims 1, 7, and 12, because the HAVi specification does not describe any mechanisms for actually implementing a class of virtual key representations. The specification only offers general guidelines. That is, the HAVi specification does not describe how virtual key information is to be stored, or where it is to be stored. Therefore, with respect to claim 1, none of the HAVi sections cited in the Office Action, whether considered independently, or as a group, describe the step of accessing a JAR file to retrieve virtual key information. With respect to claim 7, none of the cited sections describe a method of accessing a ResourceBundle to retrieve of virtual key information. With respect to claim 12, none of the cited sections describe the retrieval of virtual key information in response using

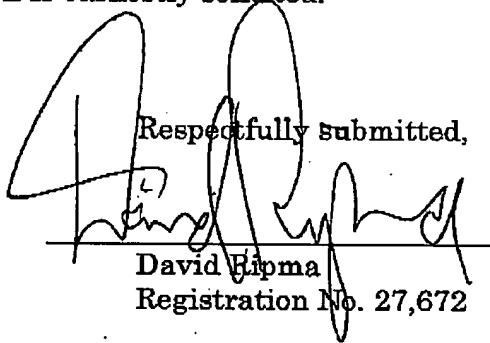
a JNI to access a mapped memory. The cited references do not explicitly describe every limitation of claims 1, 7, and 12. As a matter of well established law, a claim cannot be anticipated if the prior art reference does not explicitly describe every limitation of the claimed invention. Since the HAVi specification does not explicitly describe every limitation of claims 1, 7, and 12, it cannot anticipate. Claims 26, dependent from claim 1, claims 8-11, dependent from claim 7, and claims 13-17, dependent from claim 12, enjoy the same benefits, and the Applicant respectfully requests that the rejection be removed.

It is believed that the application is in condition for allowance and reconsideration is earnestly solicited.

Date:

2/18/05

Respectfully submitted,

David Ripma
Registration No. 27,672

Customer Number 27518
Sharp Laboratories of America, Inc.
5750 NW Pacific Rim Blvd.
Camas, WA 98607
Telephone: (360) 834-8754
Facsimile: (360) 817-8505